

Arduino와 XBee 모듈을 이용한 Delay Tolerant Network 구현 및 평가*

강인석, 김찬명, 한연희†
 한국기술교육대학교 첨단기술연구소 지능네트워크연구소
 e-mail:{iseka, cmdr, yhhan}@koreatech.ac.kr

Implementation and Evaluation of Delay Tolerant Network using Arduino and XBee module

In-Seok Kang, Chan-Myung Kim, Youn-Hee Han[†]
 Laboratory of Intelligent Networks,
 Advanced Technology Research Center

요 약

DTN (Delay Tolerant Network)은 중단 간 연결성이 보장되지 않는 환경에서의 통신을 위해 메시지 전송 방식으로 “저장 과 운반 및 전달 (Store, Carry and Forward)”을 수행하는 라우팅 방식과 불안정한 네트워크에서 메시지를 전달하기 위해 번들(Bundle)이라는 추가적인 계층을 통해 통신을 하는 네트워크이다. 본 논문은 MIT에서 개발한 Arduino라는 마이크로 컨트롤로 보드와 IEEE802.15.4를 기반으로 하는 Zigbee 통신을 가능케 하는 XBee 모듈을 이용하여 DTN을 구현한 결과를 제시하고 구현된 DTN 환경에서 Relay Node의 속도 변화에 따라 메시지 전달 시간의 변화를 보인다.

1. 서론

DTN (Delay Tolerant Network)은 통신 인프라가 충분히 갖춰지지 않은 지역이나 네트워크가 정상적으로 동작하기 힘든 지역에서 통신 지연이나 단절에 민감하지 않은 네트워크 환경을 의미한다. DTN에서의 단절과 지연은 사용자의 이동성이 존재하는 경우 빈번하게 발생하게 되는데, 이를 보완하기 위해 Bundle 계층이 추가적으로 사용된다. Bundle 계층에서는 데이터의 신뢰성 있는 교환뿐만 아니라 전송 경로가 끊길 경우를 대비하여 메시지 전송 방식을 “저장 과 운반 및 전달 (Store, Carry and Forward)” 기반으로 라우팅을 수행한다 [1].

현재 국내외에서 DTN에서 효율적인 메시지 전달을 위한 라우팅 기법을 활발하게 연구 중이지만 실제로 DTN 네트워크를 구현한 사례에 대한 발표는 많지 않다. 본 논문에서는 최근 MIT에서 개발한 Arduino라는 마이크로 컨트롤로 보드와 IEEE802.15.4를 기반으로 하는 Zigbee 통신을 가능케 하는 XBee 모듈을 이용하여 DTN을 구현한 결과를 제시하고 구현된 DTN 환경에서 Relay Node의 속도 변화에 따라 메시지 전달 시간의 변화를 보인다.

2. 관련연구

* 본 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구 (2012-00025558)이며 또한 지식경제부와 한국산업기술진흥원의 전략기술 인력양성사업으로 수행된 결과입니다.

† 한연희 : 교신저자

2.1 DTN과 DTN 라우팅

DTN은 센서 네트워크, 차량 네트워크, 로봇 네트워크와 같이 빈번한 연결성 변화로 인한 중단간 경로 부재, 에너지 소진으로 인한 네트워크 구성 변화, 높은 지연 시간 등의 특성을 지니기 때문에, 기존의 TCP/IP 프로토콜이 적용되기 힘들고 대신 기회적 메시지 전달 (Opportunistic Message Forwarding)을 기본적인 메시지 전송 방식으로 활용하는 네트워크이다. 최근 DTN과 관련하여 중단간 연결성이 보장되지 않고 각 DTN Node들의 이웃 Node들이 동적으로 변화하는 상황에서의 라우팅 기법에 대한 연구가 가장 폭넓게 이루어지고 있다. (그림 1)은 DTN에서 사용되는 Bundle Layer를 보여주고 있다. Bundle Layer는 DTN에서의 중단간 데이터 교환을 위해 제안되었으며 각각의 네트워크에서는 신뢰성 있는 데이터 전송을 보장한다.

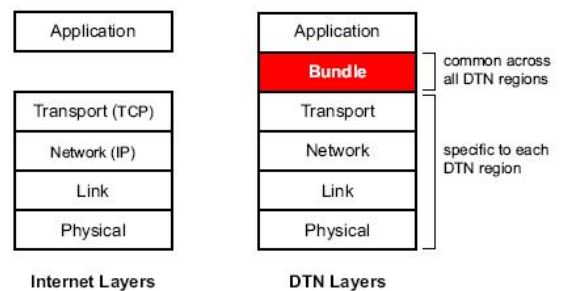


그림 1. Bundle Layer
 DTN 라우팅 방식에는 크게 ANA (Auxiliary Nodes

Assisted) 방식과 IMN (Independent Mobile Nodes) 방식으로 나누어진다. ANA 방식은 DTN에서 데이터 전송을 위해 특별한 보조의 Node들이 추가로 삽입되어 이 Node들을 이용하는 라우팅 기법이고, IMN 방식은 DTN에 어떠한 추가적인 Node의 도움없이 DTN으로 구성된 Node들에 의해서만 데이터가 전달되도록 제안하는 라우팅 기법이다. 현재 ANA 방식보다는 IMN 방식에 보다 많은 연구들이 진행되고 있으며 기존에 제안되었던 대표적인 프로토콜로는 Epidemic, Spray and Wait, Prophet 등이 있다.

Epidemic 라우팅[2]은 전염병의 확산처럼 메시지를 목적지 Node로 전달할 때까지 DTN의 모든 Node들이 메시지를 복사함으로써 전달하는 DTN의 대표적인 프로토콜이다. 위의 전략은 메시지의 전달 성공률과 도달 지연시간에서는 좋은 성능을 보이지만 각 Node당 메시지 전달 오버헤드가 크다는 단점이 있다. Spray and Wait[3]는 메시지마다 복사할 수 있는 수를 제한하여 전송하는 전략이고, PROPHET[4]은 각 Node들이 서로간의 접촉 (Contact) 정보를 유지하고 이를 통해 메시지가 목적지에 전달될 가능성이 자신보다 큰 Node에게만 메시지를 전달함으로써 무분별한 메시지 전달 횟수를 줄인다.

2.2 Arduino

Arduino[5]는 MIT에서 개발되었으며 Open Source를 기반으로 한 피지컬 컴퓨팅 (Physical Computing) 플랫폼으로 AVR을 기반으로 한 보드와 소프트웨어 개발을 위한 통합 환경을 제공한다. Arduino의 큰 장점으로는 마이크로 컨트롤러를 쉽게 동작시킬 수 있다는 것이다. 마이크로 컨트롤러 프로그래밍을 위해서는 일반적으로 AVR 프로그래밍을 WinAVR로 컴파일하여 ISP 장치를 통해 업로드를 해야 하는 등 번거로운 과정을 거쳐야 하는데 Arduino는 오픈소스 통합개발환경 (IDE)을 통해 비교적 간단한 Processing언어로 프로그래밍 및 컴파일하여 USB를 통해 업로드를 쉽게 할 수 있다. 또한 다른 모듈에 비해 비교적 저렴하며 여러 OS를 지원한다.

2.3 XBee

본 논문에서 구현하는 DTN에서 사용되는 통신 모듈은 IEEE802.15.4 기반의 Zigbee 프로토콜을 지원하는 Digif[6]사의 XBee 모듈 Series1이다. 본 논문에서는 DTN의 Bundle을 Application 계층에서 구현하였다.

XBee 통신 모드에는 AT 모드와 API 모드 두 가지가 있으며, 본 시스템에서는 API 모드를 이용하여 구현하였으며, Arduino에서 XBee 통신을 위해 개발된 XBee API for Arduino[7]를 사용하였다.

3. 시스템 구조

(그림 2)는 본 논문에서 구현한 DTN Node 구성 요소를 보여준다. 각 구성 요소로는 Arduino, XBee 쉴드

(shield), XBee 모듈, LEGO Mindstorm이 사용되었으며 XBee 모듈 통신은 XBee API for Arduino를 이용하여 구현하였다. Arduino 및 XBee 모듈, XBee 쉴드는 하나의 몸체로 결합하였고 LEGO Mindstorm과도 결합하여 각 Node가 스스로 구동하여 이동할 수 있도록 하였다. <표 1>은 각 구성요소의 사진을 보여준다.

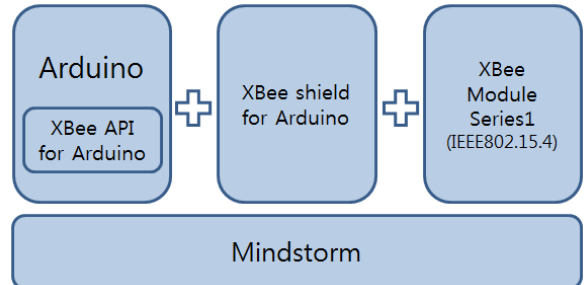






그림 2. 시스템 구조

<표 1> Node 구성요소

	
Arduino UNO R3	XBee module Series1(IEEE802.15.4)
	
XBee shield	Mindstorm NXT2.0

3.1 Bundle (Data) Frame 구조

본 논문에서 구현하는 DTN에서 활용되는 통신 방식에서는 (그림 3)과 같은 데이터 구조를 사용하였다. 총 5개의 필드로 구성되었으며 Destination Address 필드는 메시지의 최종 목적지 주소를 담고 있으며, Source Address는 최초 메시지를 전달한 Node의 주소를 담고 있다. Previous Address는 메시지가 전달된 바로 직전의 Node의 주소를 담고 있으며 이러한 Previous Address 필드를 이용하여 메시지의 불필요한 재전송을 막을 수 있다. Payload 필드는 전달되어야 하는 데이터를 담고 있으며, Payload ID 필드는 각 Payload를 판별하는 고유한 ID 값을 담고 있다.

Destination Address	Source Address	Previous Address	Payload ID	Payload
---------------------	----------------	------------------	------------	---------

그림 3. Bundle Frame

3.2 DTN에서 각 Node의 기본 기능

- Forward

자신이 전달하여야 하는 메시지를 Bundle 계층에 저장 (Store)하고 있는 경우, 새롭게 통신 가능한 이웃 Node가 발견되면 그 Node로 해당 메시지를 전송한다. 이 때, Payload ID를 활용하여 메시지를 수신하는 Node가 이전에 그 메시지를 수신한 적이 있다면 그 메시지를 해당 Node에게 전송하지 않는다.

● Receive

다른 Node로부터 전송된 메시지를 수신하여 Bundle 계층에 저장한다.

● Store

다른 Node로부터 수신된 메시지를 Bundle 계층에 저장한다.

● Carry

다른 Node로부터 전달된 메시지를 저장한 상태에서 이동을 한다.

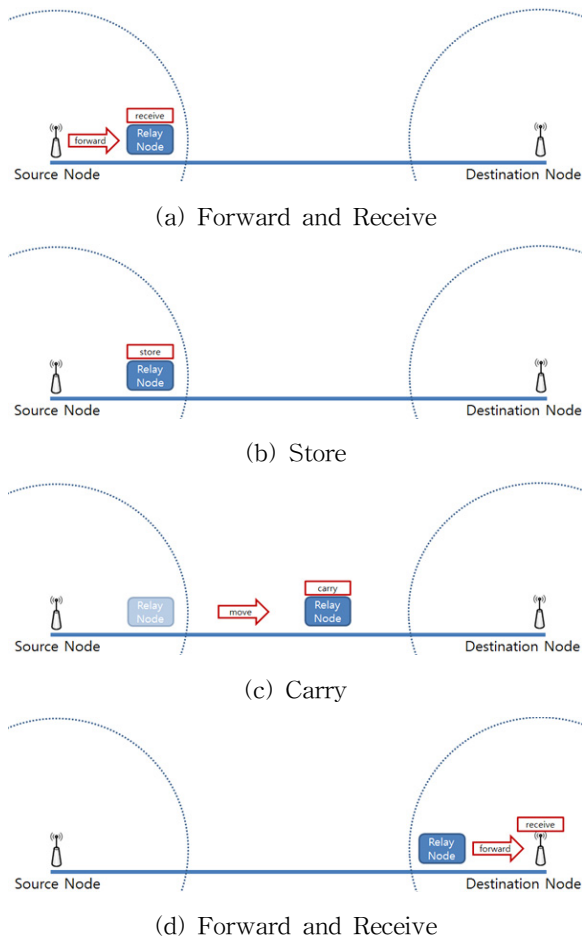


그림 4. DTN Node의 기본 기능

(그림 4)는 DTN Node의 기본 기능을 활용하여 기본적인 메시지 전달과정을 도시하고 있으며, 그림에서 보듯이 Source Node에서 Destination Node까지 메시지의 전달은 중간에 위치한 Relay Node를 통해 이루어진다.

각 Node는 자신의 존재를 알리는 Beacon 메시지를 브로드캐스트한다. 임의의 Node는 다른 Node로부터 Beacon Message를 받으면 그 Node를 이웃(Neighbor) Node로 설

정한다. Beacon Message로 인해 통신이 가능한 이웃 Node와 접촉이 이루어지면 해당 Node는 그 이웃 Node에 Bundle을 전달할지 판단한다.

4. 구현 및 평가

본 DTN 시스템의 실험 환경은 (그림 5)와 같다. Source Node와 Destination Node와의 직선거리는 약 1m이며 Source Node와 Destination Node사이에서 일정하게 이동을 하는 Relay Node로 구성되어 있다. 총 사용되는 Node는 3대의 Node이며 Source Node에서 Destination Node로의 Bundle Data의 전달을 목적으로 한다.

각 XBee 모듈의 통신 반경은 실내 환경에서 실제로는 최대 약 30m이지만 본 실험에서는 한정된 1~2m 지역 내에서 실험환경을 구축하기 때문에 RSSI 임계값 (γ)을 활용하여 각 XBee 모듈의 통신 반경을 약 25cm로 줄였다. 즉, 임의의 Node가 다른 Node에 대해 다음과 같은 조건을 만족할 때 통신이 가능하다고 간주하며 그 Node 들끼리는 서로 이웃 Node가 된다.

$$RSSI < \gamma$$

위 식에서 γ 는 실험 환경에서 경험적으로 (heuristically) 정하였다. 마지막으로 모든 실험에서 각 Node의 Beacon 주기는 1초로 설정하였다.

Relay Node는 Source Node와 Destination Node와의 통신 이루어지는 지점을 반복적으로 움직이며 이동 궤적은 그림 4에 표시되어 있다. 즉, Relay Node는 이동을 하면서 주기적으로 Source Node 및 Destination Node와 이웃 Node 관계를 교대로 설정하게 된다.



그림 5. 실험환경

본 DTN 시스템에는 XBee API for Arduino를 이용하여 직접 DTN에서 사용되는 Bundle 구조를 구현하였다. Bundle 구조를 Arduino에 구현하기 위하여 "DtnNode 클래스"를 구현하였다. 다음은 DtnNode 클래스의 주요 멤버 변수 및 함수 목록 설명이다.

[멤버 변수]

- unsigned long beaconInterval : Beacon Message 전송 주기
- uint8_t bundle[BUNDLE_SIZE] : Bundle Frame
- uint8_t bundleStore[MAX_NUM_BUNDLES][BUNDLE_SIZE] : Bundle 저장소
- uint16_t neighborAddress[NEIGHBOR_NUM]

: 이웃 Node 주소 리스트

[멤버 함수]

- void setBeaconInterval(unsigned long interval)
: Beacon Message 전송 주기 설정
- void setBundle(uint8_t destination, uint8_t source, uint8_t previous, uint8_t data, uint8_t dataID)
: Bundle Frame 설정
- Rx16Response& getRxBundle()
: Bundle Frame 수신
- bool checkBundle(uint16_t addr)
: Bundle Frame 전송할 Node의 유효성 판단
- void addNeighborAddress(uint16_t addr);
: 이웃 Node 리스트에 추가

(그림 6)은 본 실험을 위하여 구현한 DTN Node 모습이다.



그림 6. DTN Node

본 실험의 목적은 Relay Node의 이동 속도 변화에 따른 메시지를 전송 받는 지연 시간을 평가하는 것이다. Relay Node가 주어진 궤적을 따라 주기적으로 이동을 할 때, 전송 지연 시간(Message Delivery Latency)은 Source Node에서 첫 번째 메시지를 Relay Node에게 전송한 시각부터 Relay Node가 그 메시지를 Destination Node에 전달하고 다시 Source Node로 이동하여 Source Node가 두 번째 메시지를 Relay Node에게 전송한 시각까지를 측정하였다.

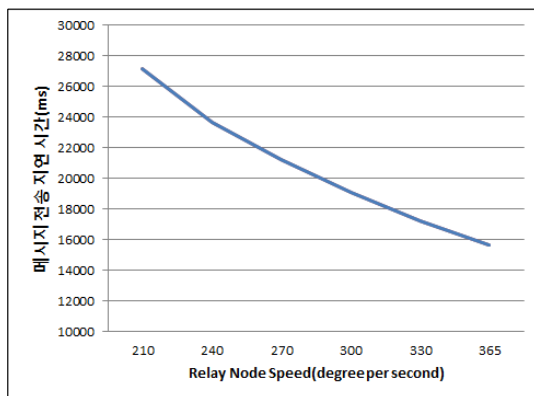


그림 7. Relay Node의 속도 변화에 따른 메시지 전송 지연 시간

(그림 7)은 Relay Node의 속도 변화 (초당 회전각 변화)에 따른 메시지 전송 지연 시간 변화를 보여준다. 실험 결과를 얻기 위해 임의의 Relay Node 속도 설정에 대해 총 20회의 실험을 통하여 산출된 값의 평균값을 활용하였다. 실험 환경에서 일정하게 움직이는 Relay Node의 속도가 증가하면 메시지 전송 지연 시간은 그에 따라 감소하였다. 그 이유는 이동하는 Relay Node의 이동 속도가 증가하면 그 만큼 Source Node와 Destination Node에 접근하는 시간 간격이 줄어들어 메시지를 수신, 저장, 전달하는 시간 또한 줄어들기 때문이다.

5. 결론

본 논문은 Arduino와 XBee 모듈을 이용하여 DTN을 구현한 결과를 제시하고 구현된 DTN 환경에서 메시지 전달 과정의 Relay Node의 속도 변화에 따라 메시지 전달 시간의 변화를 보였다. 향후에는 실험 환경에 보다 많은 DTN Node들을 배치하고 임의의 Node가 Source 및 Destination이 되고 다수의 Node가 Relay의 역할을 수행하도록 할 예정이다. 또한, 이러한 실험 환경에서 Epidemic, Spray and Wait, Prophet 등의 라우팅 방식을 구현하여 각 방식의 장단점을 비교 평가하고 Social Network의 특성[8]을 적용하여 보다 효과적인 라우팅 기법을 제안 및 평가할 예정이다.

참고문헌

- [1] Delay Tolerant Networks (DTNs), A Tutorial, 2003 <http://www.dtnrg.org>
- [2] A. Vahdat and D. Becker, Epidemic routing for partially connected ad hoc networks, Technical report CS-200006, Duke University, Apr. 2000.
- [3] Thrasymoulos Spyropoulos, Konstantinos Psounis, Cauligi S. Raghavendra. Spray and wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. SIGCOMM 2005.
- [4] Tara Small et al. Probabilistic Routing in Intermittently Connected Networks. SIGCOMM'05 Workshops, August 2005.
- [5] <http://www.arduino.cc/>
- [6] <http://www.digi.com/>
- [7] <http://code.google.com/p/xbec-arduino/>
- [8] 이규행, 박태민, 김종권, 소셜 커뮤니티 기반의 DTN 라우팅, 정보과학회논문지 2011년 10월